

浅谈前端工程化

美团点评 前端工程师 何骏

2016-03-21



个人简介

- 12年毕业加入腾讯， 15年加入点评
- 参与过webQQ， Q+， QQ互联， 手机QQ， 点评帐号， node server框架
- 目前主要技术方向
 - Hybrid模式移动web开发
 - 前端性能优化
 - nodejs



主要内容

- 如何理解工程化？它能给我们带来什么？
- 项目中遇到的问题
- 如何一步一步解决？
- 一些思考

工程化

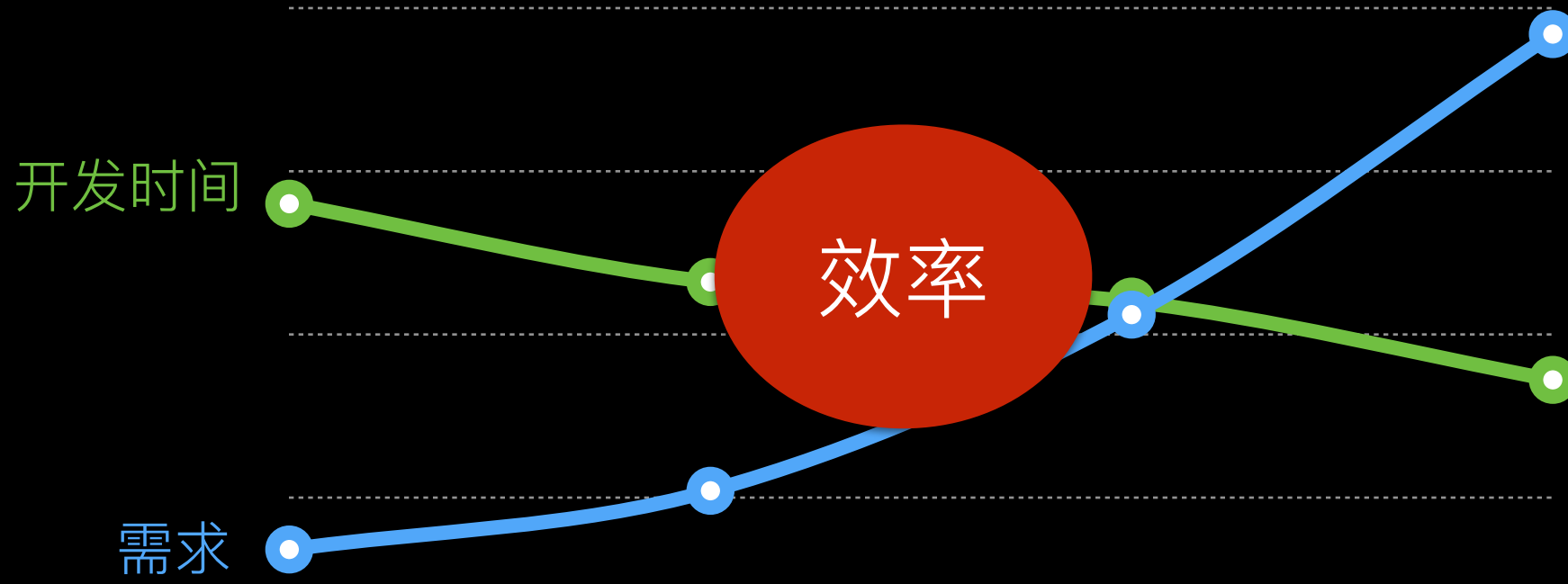
- 什么是工程化？

工程化就是将流程、技术、工具、经验等规范化、标准化

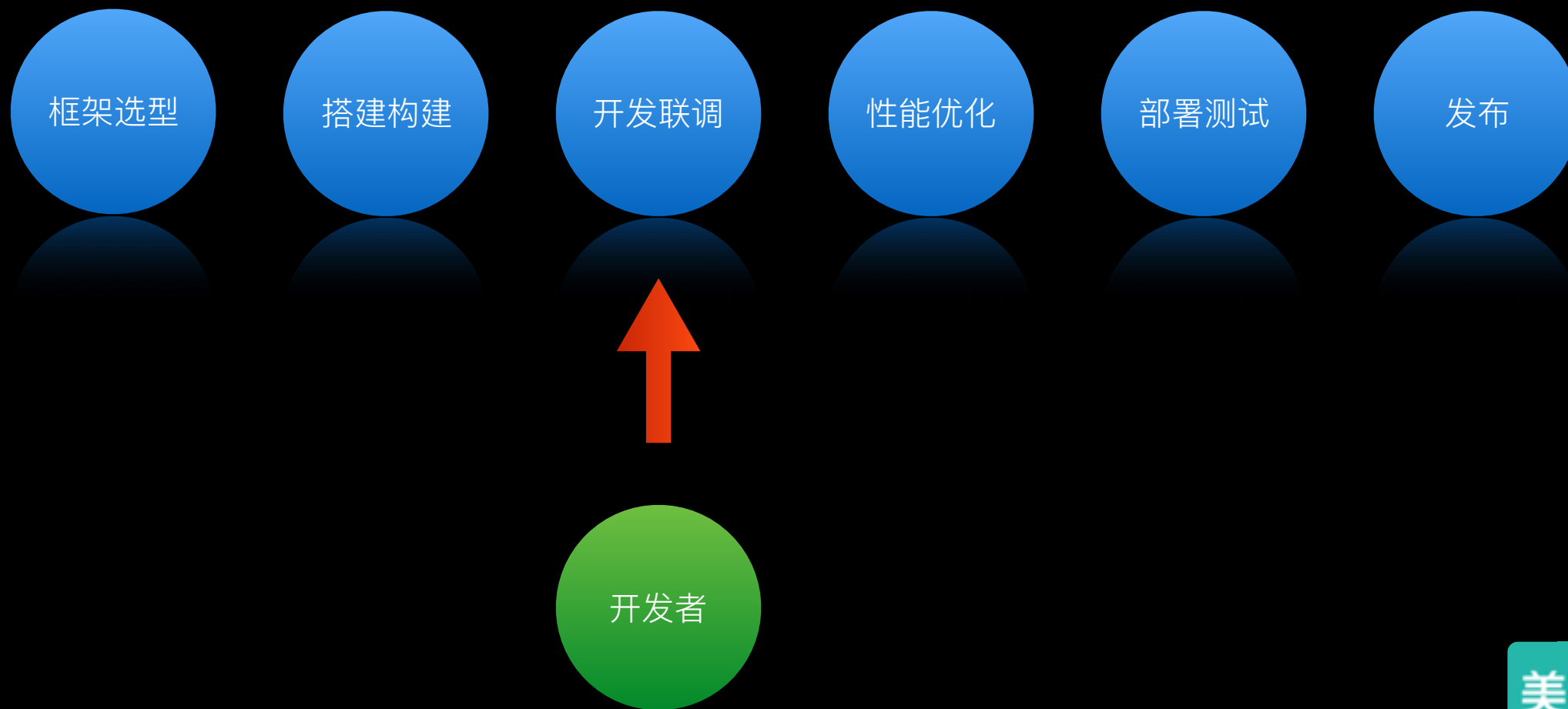
- 为什么需要工程化？

前端涉及领域越来越广，项目复杂度越来越高

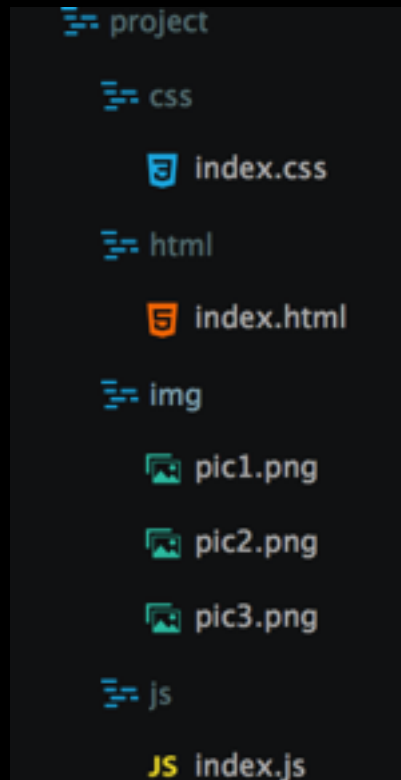
工程化



工程化



遇到的问题

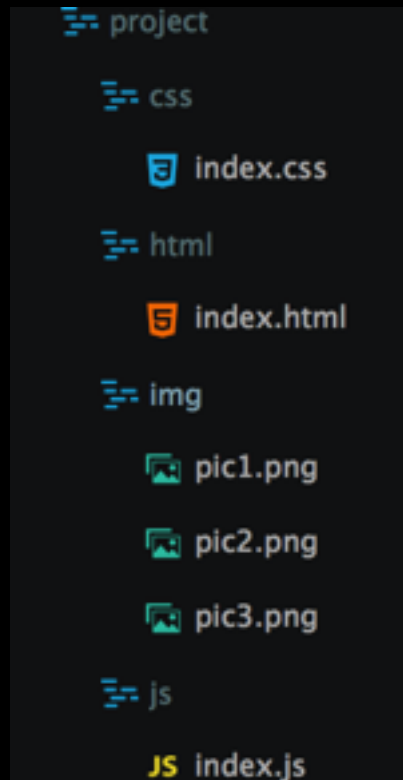


ftp

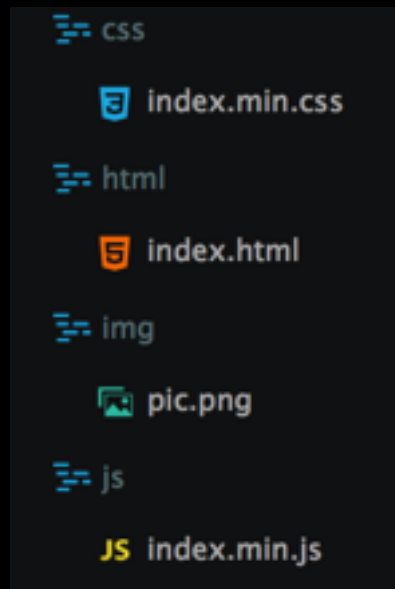


server

遇到的问题



构建

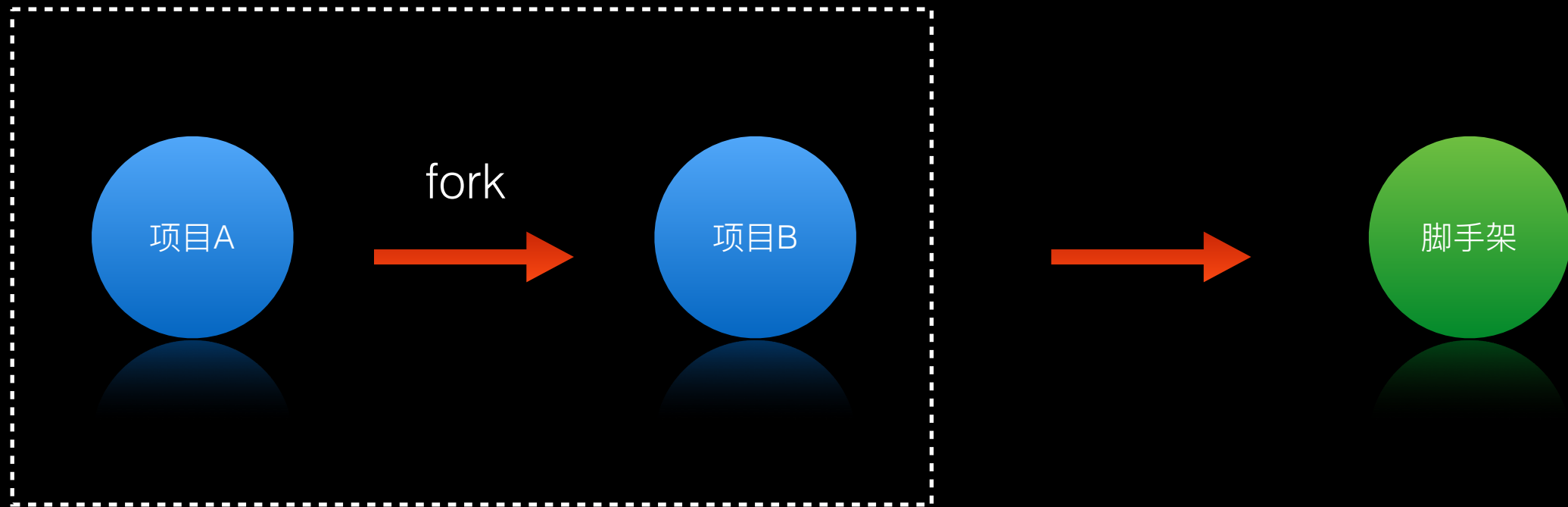


ftp



server

遇到的问题



如何解决？

用yeoman作为项目脚手架



- ✓ 简单易用，门槛低
- ✓ 脚手架自身作为npm包，可复用，易改良
- ✓ 由专人持续维护，利于团队协作和统一规范

如何解决?

node-server脚手架

```
phantomdeMacBook-Pro:node-server-test phantom$ yo
? 'Allo phantom! What would you like to do? (Use arrow keys)
  Run a generator
> @dp/node-server
  React
  Webapp
  React
  Update
(Move up)

Make sure you are in the directory you want to scaffold into.
This generator can also be run with: yo @dp/node-server

This utility will walk you through creating a project.
? project name node-server-test
? version 0.1.0
? description node server
? git repository git@code.dianpingoa.com:phantom.he/node-server-test.git
? keywords node server
? author (phantom.he)
```

```
node-server-test
├── action
├── config
├── node_modules
├── view
├── .gitignore
├── JS app.js
├── package.json
└── README.md
```

遇到的问题

模块化

```
function m1() {  
    ...  
}  
  
function m2() {  
    ...  
}
```

```
var module = {  
    x: 1,  
    m1: function() {  
        ...  
    },  
    m2: function() {  
        ...  
    }  
};
```

```
var module = (function() {  
    var x = 1,  
        m1 = function() {  
            ...  
        },  
        m2 = function() {  
            ...  
        };  
  
    return {  
        m1: m1,  
        m2: m2  
    };  
})();
```

遇到的问题

规范化？ 依赖关系？

The logo for CommonJS, featuring the word "Common" in a brown sans-serif font and "JS" in a larger, stylized brown font where the "J" and "S" are connected.The logo for Node.js, with "node" in a white lowercase sans-serif font and ".js" in a green hexagonal icon.The logo for Browserify, featuring the word "browserify" in a stylized, gothic-like font with a blue flame-like effect on the "y".

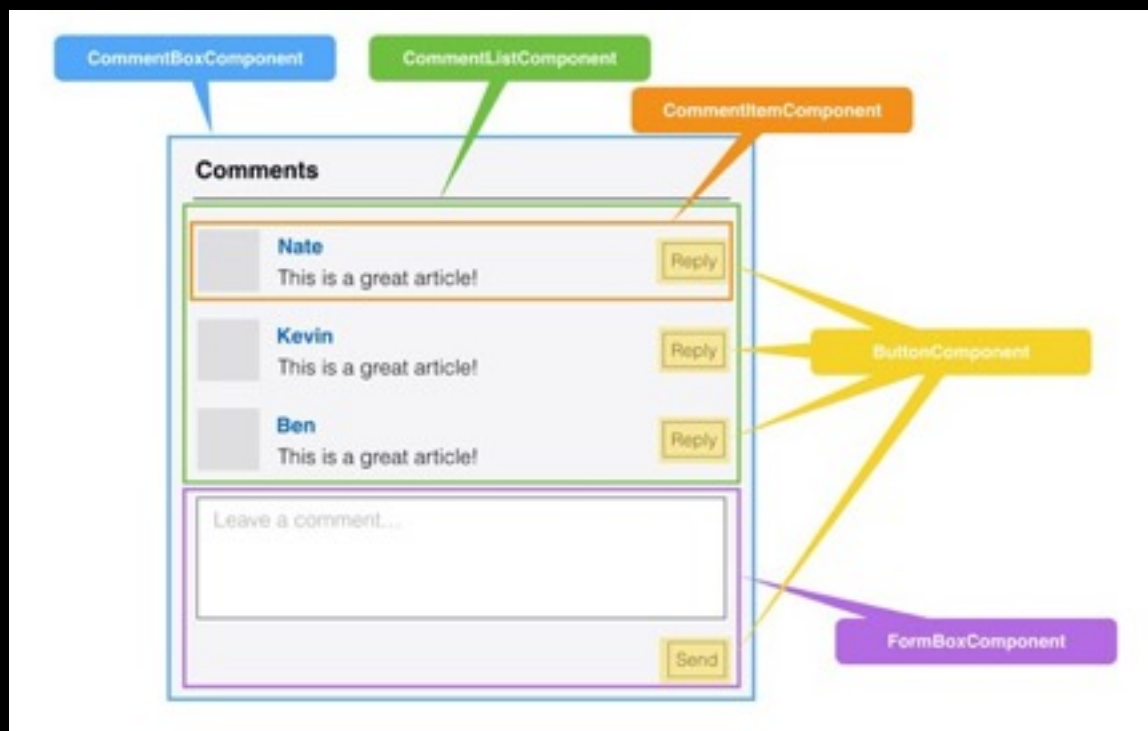
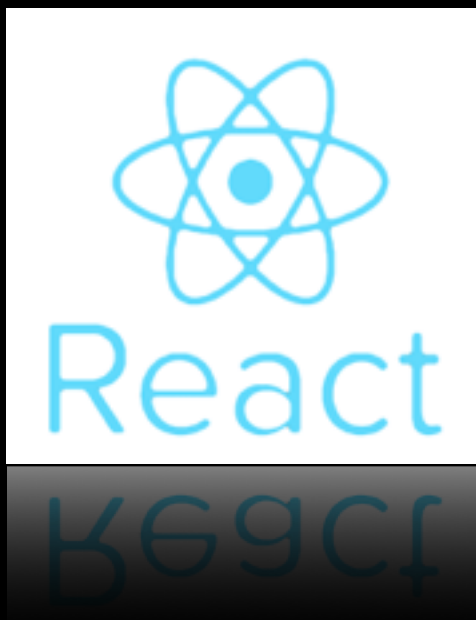
webpack
MODULE BUNDLER

美团
meituan.com



遇到的问题

组件化



✓ 可复用

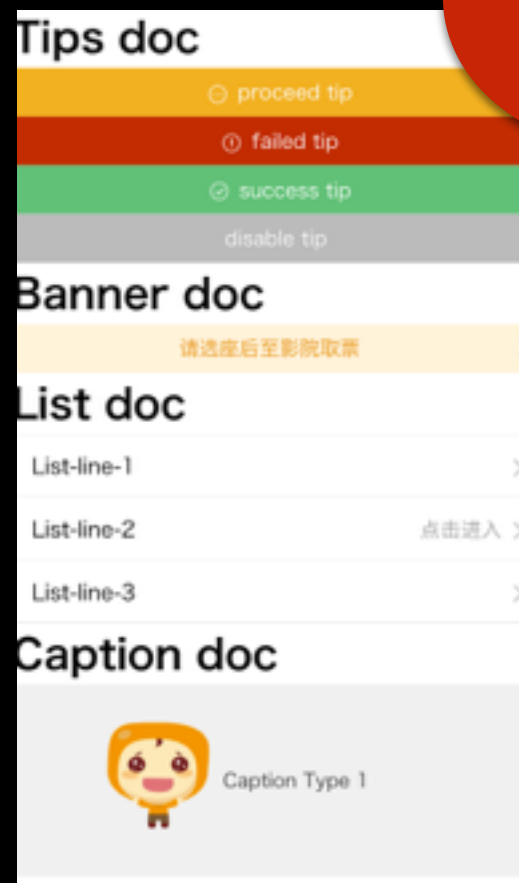
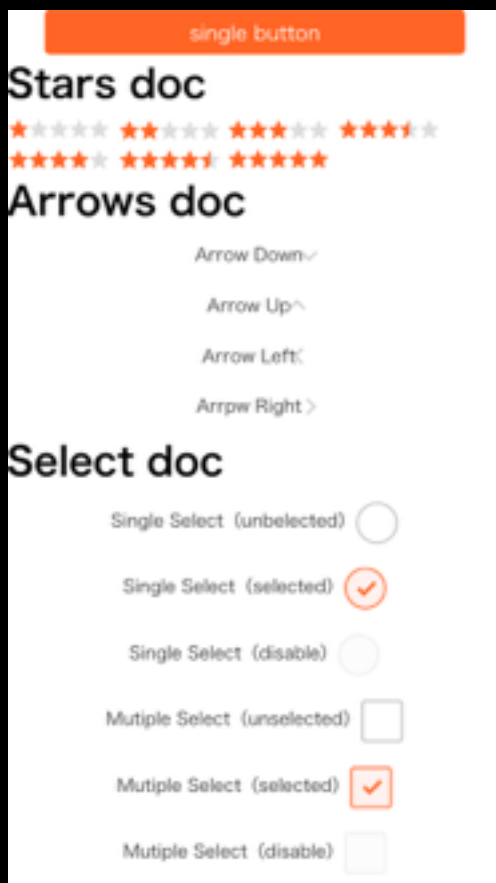
✓ 可组合

✓ 易维护

如何解决? —— 开发效率

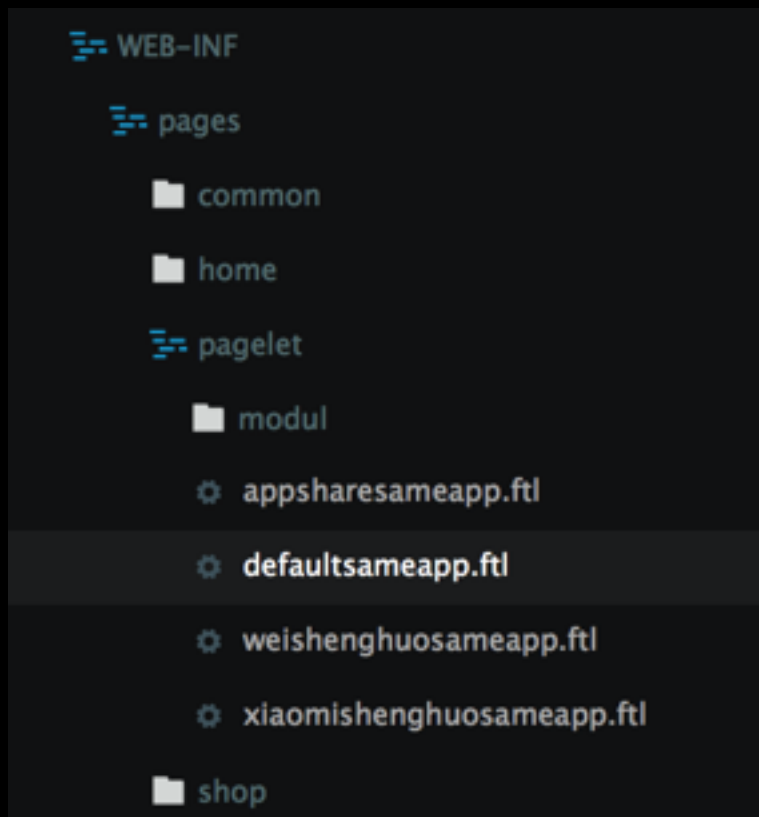
模块化、组件化

100+



遇到的问题

模板的维护



```
<#--加载模块-->
<@cortex.facade mod="hippo"></@cortex.facade>
<@cortex.facade mod="nugget-mobile"></@cortex.facade>

<@cortex.facade mod="app-m-freshman/js/show-gift">
</@cortex.facade>

<@cortex.facade mod="app-m-dianping-shop">
{
    shopId: '${shopId}',
    regionId: '${shopVo.shopDTO.regionId}',
    categoryId: '${shopVo.shopDTO.categoryId}',
    title: '${shopVo.shopDTO.shopName}<#if shopVo.shopDTO.branchName??>(${shopVo.
shopDTO.branchName})</#if>',
    desc: 'test',
    image: '${shopVo.shopDTO.defaultPic}'
}
</@cortex.facade>
```

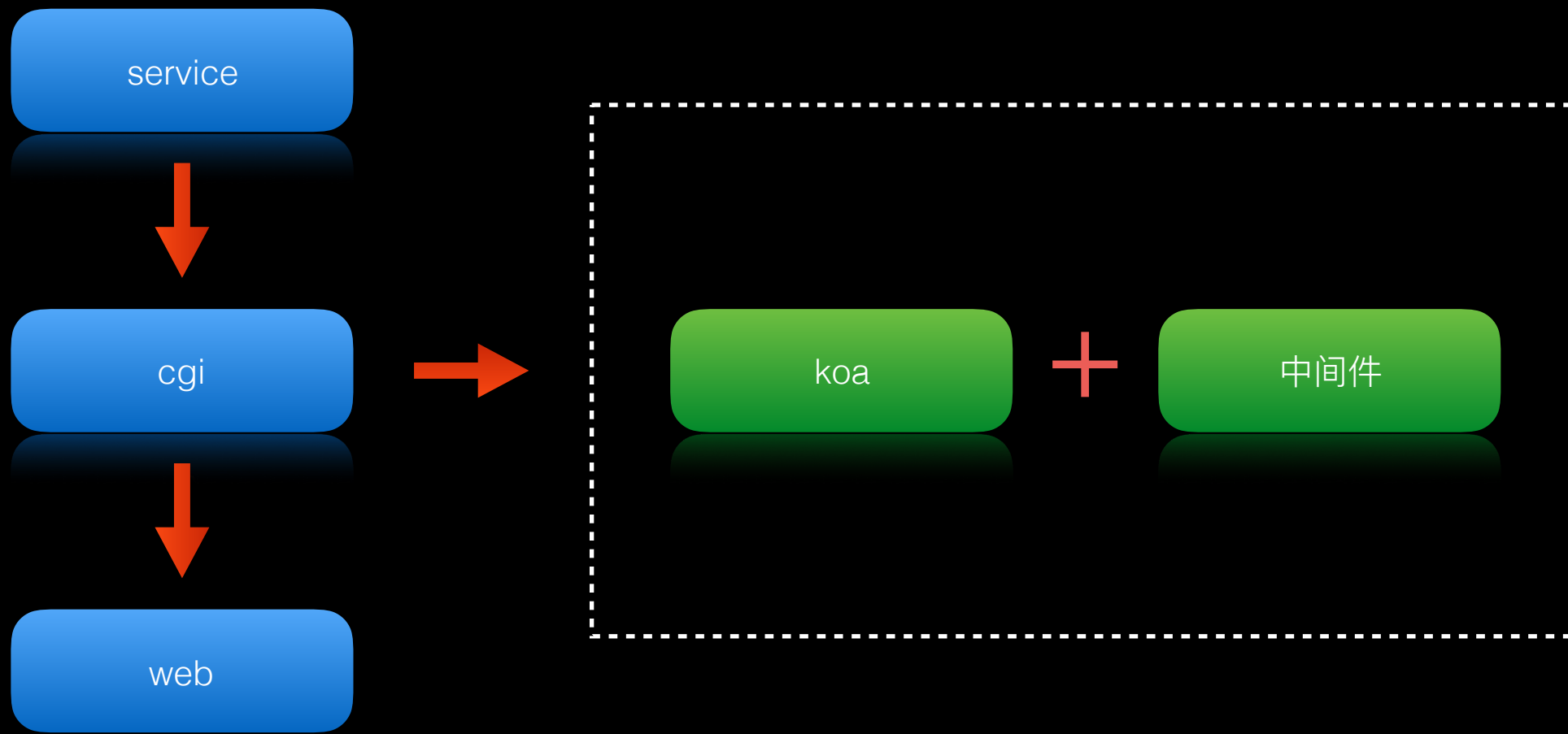

遇到的问题

接口联调

- 你返回的格式和约定的不一样啊?
- 能不能给我加一个参数?
- 排序的逻辑应该放在你这边吧?
- 支持一下jsonp吧?
-



如何解决? —— 前后端分离



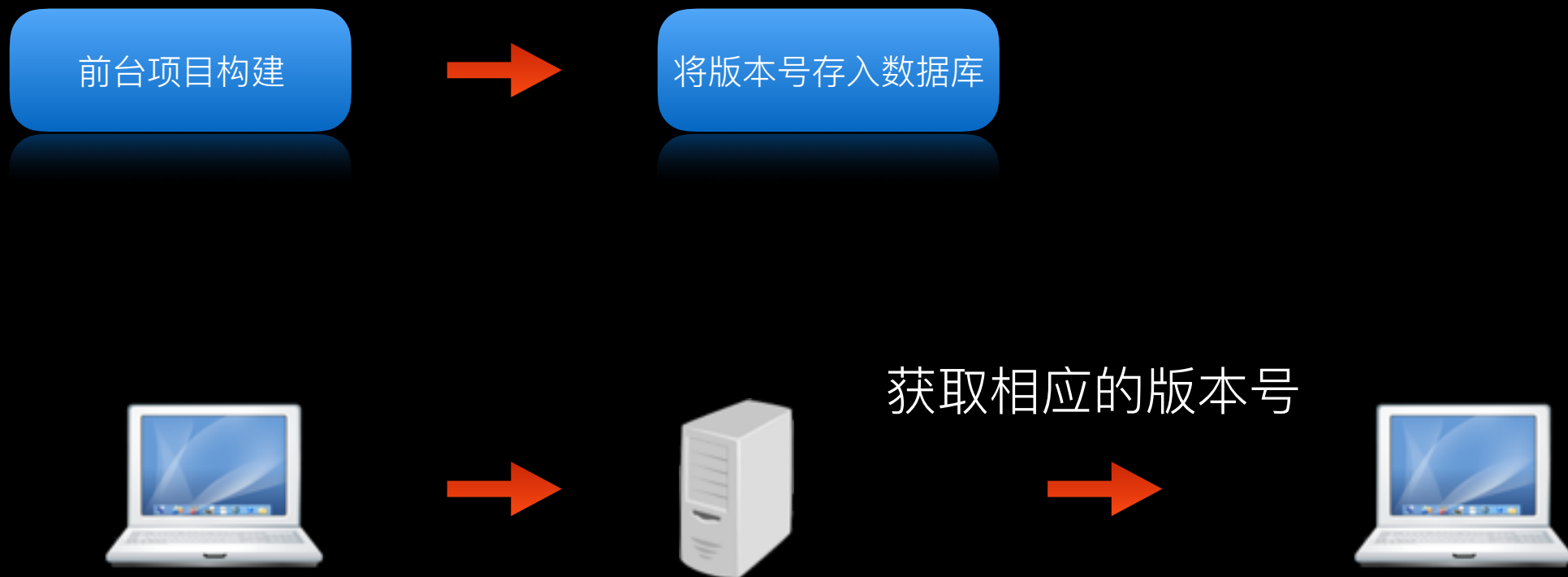
如何解决? —— 前后端分离

- ✓ 职责划分更清晰
- ✓ 有效减少了沟通和开发的成本
- ✓ 借助React实现同构



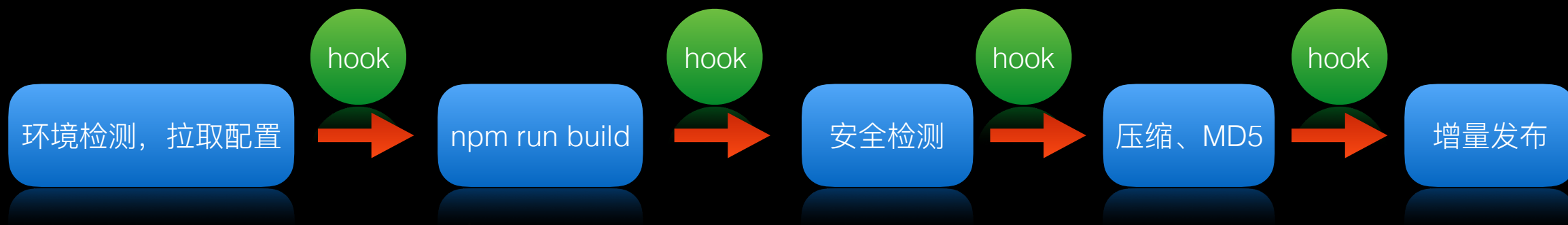
遇到的问题

以前的构建



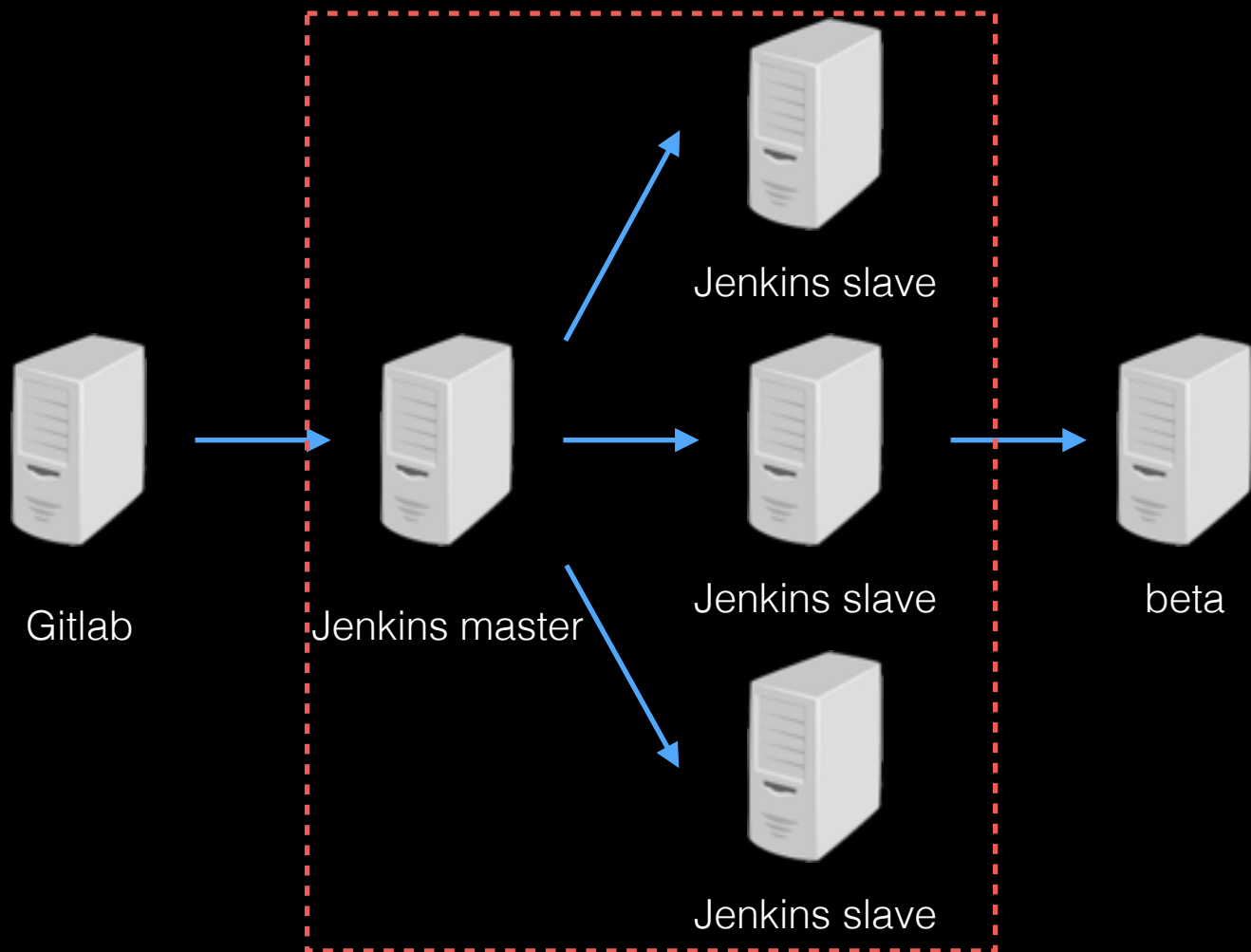
如何解决? —— CI

主要流程



如何解决? —— CI

✓ 通过负载均衡加快构建速度



如何解决? —— CI

✓ 灵活可配的任务队列 + 钩子

×

编辑配置项[cortex-ci.peon.task.config]

类型

string

trim

环境:

☐ 全选线下

☐ 全选线上

☐ 开发

☐ Alpha

☒ Beta

☐ 预发

☐

线上-南汇机房

☐ 性能

Value:

```
{
  "static":["env", "npm","clean","build",
    "file_check","cp","imagemin","css_bg","cssmin","jsmin",
    "md5","html","ftp","dbversion" ],
  "web":["env","npm","web_zip"],
  "pkg":["env","npm","npm_publish"]
}
```

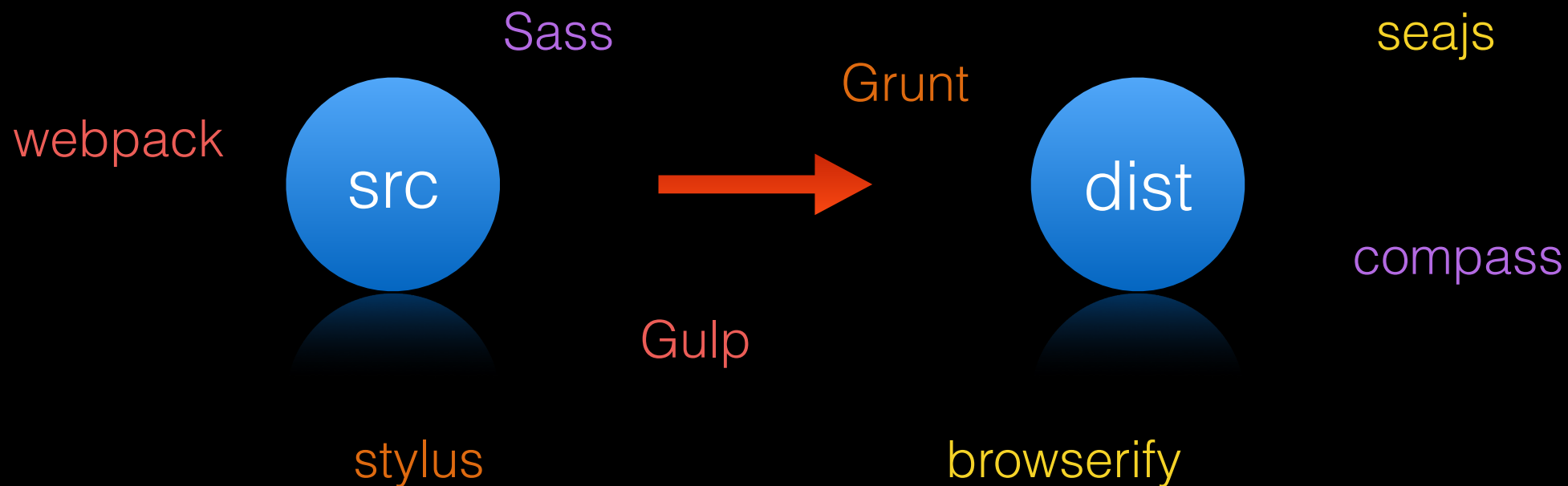
关闭

保存

编辑泳道配置

如何解决? —— CI

✓ 将build过程交给项目自身，达到解耦的目的



如何解决? —— CI

- ✓ 对发布前的代码统一进行xss等安全扫描, 有效保证了代码的质量
- ✓ 增量发布, 进一步缩短了发布的时间



不足之处

- 调试工具（移动端）
- CI加入自动化测试（多终端）
- 团队规范的执行



一些思考

- 不要总是瞎忙
- 项目初期可以怎么快怎么来，但后期还是要一劳永逸
- 多总结多沉淀，温故而知新

Q&A

Thank you!

